

# **Protocol for WEB API for Members**

## **NCMS (CM Segment)**

Version 1.0



The NSE Clearing Limited (National Clearing)  
Exchange Plaza, Plot No. C/1, G Block,  
Bandra-Kurla Complex, Bandra (E),  
Mumbai - 400 051

## **NSE Clearing Confidential**

### **Notice**

© Copyright NSE Clearing Ltd (NCL). All rights reserved. Unpublished rights reserved under applicable copyright and trades secret laws.

The contents, ideas and concepts presented herein are proprietary and confidential.  
Duplication and disclosure to others in whole, or in part is prohibited

---

**Revision History**

<b>Date</b>	<b>Change Description</b>	<b>Edited By</b>	<b>Version</b>
29-Oct-2025	Initial version		1.0

Confidential

---

## Table of Contents

Revision History .....	2
Introduction .....	4
General Instructions.....	4
HTTP Status Codes .....	4
Common Error Response JSON .....	5
Environment Details.....	5
API Consumer Registration .....	5
API Security .....	6
Clearing Corporation APIs .....	7
POST /<version>/request/token .....	7
POST /<version>/inquire/trd-act .....	9
Appendix A - Reference Codes.....	14
Market Type .....	14
Market Status.....	14
Transaction Code .....	14
Activity Type.....	14
Book Type.....	14
Client Type .....	15
Buy Sell Flag .....	15
Trade Status .....	15
Exchange Code .....	15
Async Response Code .....	15

Confidential

## Introduction

This document provides information on the Web APIs used for programmatic access to trade management related data between NCL's NCMS Platform and its Members. It details the messaging protocols and structures required to develop this interface.

## General Instructions

1. Following headers need to be provided in all API calls made to clearing corporation.
  - **Content-Type:** This header should be provided in all requests with method as "POST". Header value should be "application/json".
  - **User-Agent:** All requests should contain this header. The value of "User-Agent" header can be "/".
  - **Accept-Encoding:** This header is required in all API calls to CC. The value of this header should be blank.
  - **Accept:** This header value should be "application/json"
2. Some of the key specifications related to JSON and standards followed for the API's are as follows
  - JSON is built on 2 structures. Map containing key value pairs and an ordered list of values.
  - A value could be boolean (true / false), number, decimal, String or a structure (List or Object).
  - Object or key value pair structure consists of keys which are strings and values of any of the above types. E.g. {"name": "Amit", "age": 25}
  - List contains list of values. E.g. ["Amit", "Ajay", "Vikas"]
  - A Boolean has only 2 values true or false.
  - String values are enclosed in double quotes. e.g. "name", "Amit", "Pending"
  - Numbers and decimals are represented without any thousand - separator character. Decimal indicator is dot (".")
  - Numbers have an optional maximum number of digits. If not specified, then it is defaulted to 18.
  - Decimals have 2 mandatory length parameters. The first length parameter indicates number of digits in the whole part (before decimal place) and the second length parameter indicates number of digits in the decimal part (after decimal place).
3. All URLs for API will be always in lower case.
4. All JSON field names will follow camel-hump style of naming. A field with multiple words would be concatenated without spaces. All characters will be in lower case. First characters of words other than the first word in the field name will be in upper case. For e.g. field for "Order Number" could be represented by field name "orderNumber". Other examples are "firstName", "lastName".
5. In case of JSONs representing an object or a key-value pair, keys with null values could be omitted from the JSON.

## HTTP Status Codes

All API's will respond with an HTTP status code. A status code of 200 would indicate successful execution of the API and the response body would be as defined in the API specification.

Confidential

In case of an error a HTTP status code other than 200 will be returned. The API may or may not return an error response JSON depending upon the type of error encountered. Following are the HTTP status codes that could be returned by the APIs

#	Status Code	Reason	Description
1	200	SUCCESS	Request was handled successfully
2	400	BAD REQUEST	Indicates a validation / business logic error / json parsing errors
3	401	UNAUTHORIZED: Failed to authenticate the request	Indicates that the credentials / access token shared for authentication is invalid or expired.
4	404	NOT FOUND	Incorrect URL or Resource does not exist
5	405	METHOD_NOT_ALLOWED	Unsupported HTTP Method: A request was made for a resource using a request method not supported by that resource (e.g. using GET instead of POST).
6	500	UNKNOWN_ERROR	Internal Server Error. Such errors are to be reported to the support desk.
7	503	SVC_UNAVAILABLE	Service unavailable.

#### **Common Error Response JSON**

Field	Type	Mandatory	Description
code	Number	Yes	Http Status Code. See above
messages	List<String>	Yes	One or more error messages

#### **Sample Response**

```
{
  "code": 400,
  "messages": [
    "Invalid JSON."
  ]
}
```

#### **Environment Details**

Base URL for all CM Segment Trade Management API endpoints mentioned in this document will be as follows:

Testing Environment: <https://uat.connect2nsccl.com/CM/TRD/>

Live Environment: <https://www.connect2nsccl.com/CM/TRD/>

#### **API Consumer Registration**

To initiate data consumption through the API endpoints, members are required to submit necessary information, including their IP address and registered email address, to NCL. Additionally, members must provide their public key certificates to NCL to enable payload encryption. The public key should be generated using the RSA algorithm and comply with the X.509 standard to ensure compatibility. Once this information is received, the member will be registered for API access and provided with a Consumer Key and Secret.

Confidential

***API Security***

OAuth 2.0, an industry-standard authorisation protocol, is employed to facilitate access to API endpoints. Members can generate bearer tokens through the designated API call (refer to details below). The token response payload's data field will be asymmetrically encrypted using the Member's Public Key Certificate with the RSA algorithm. This encrypted payload will be delivered as a Base64-encoded string.

Furthermore, an AES secret key and IV unique to the member will be included within the access token payload and retained by both NCL and the member. This will serve to enable secure encryption and decryption of API payloads.

Confidential

---

## Clearing Corporation APIs

This chapter gives details of the API's exposed by clearing corporation and to be consumed by members.

### **POST /<version>/request/token**

To obtain a token, the member's consumer app must request for the access token using API POST /<version>/request/token endpoint. The access token can be reused to access NCL API data until it expires (after 'n' minutes). During API registration, the member receives a consumer key and secret, which are validated for token authorization. The access token payload also contains aes\_secret\_key and aes\_iv required to decrypt response payloads.

### **Request**

Get Token Request Header Parameters

#	Parameter Name	Data Type	Description	Sample Value
1	Authorization	String	The format should be as follows: Basic <member_credentials>  Here, member_credentials refers to a base64-encoded string consisting of the following data: cons_key:cons_secret	Basic MRZmwzCl6.....SGq ICaxH9rAM3hVIMJzFg==
2	nonce	String	A nonce uniquely identifies each server request. It should be a base64-encoded string in the format: ddMMyyyyHHmmssSSS:<6-digit random number>.	MjAwMTlwMTcxNjEyMjE0TE6
3	grant_type	String	Value MUST be set to "client_credentials".	client_credentials

### Sample Request

```
POST /auth/token HTTP/1.1
Host: www.connect2nsccl.com
Content-Type: application/x-www-form-urlencoded
Authorization: Basic MRZmwzCl6.....SGqICaxH9rAM3hVIMJzFg==
nonce: MjAwMTlwMTcxNjEyMjE0TE6ODk0MjY3
x-www-form-urlencoded
grant_type=client_credentials
```

### **Response**

The response's data field includes the encrypted token response payload as a Base64-encoded string. To access the raw token payload, first decode the Base64 data string, then decrypt the resulting bytes using the Member private key associated with the public certificate provided during the API Consumer Registration process.

### Success Response Sample

```
HTTP/1.1 200 OK
Content-Type: application/json
```

Confidential

```
{
  "data":
  "GHZovnrYUaw6X8J9GE1vfLLlgb6b/KIVp6B0uKttHP91FIFNEpEZIMI43eWMcyOUEsvqr5fj4snHA125K8++8U/R
tCYC7r3bW+2U/P6J/nG2qNtFGRoM1Koc0KVMcFgNptJC6BK2Bs6Fo44KAOtJ97NBIf9R0/WPxJy3dqi2A6zXo9t
qn22JfgaFq/2JWZT0kX1grGkBEJZZImUiA0+ftpV3JfqrYwZAtCr+cM7nbhab8Mri8cWBeHNG1pALU/A1jcDvar5
/NTdMDSCImkuw7ngXQpnOFX1mP1AITAYLHOTnuau3KoE653lze2+ruleMuk9celEuL+vahYqtZfz7w=="
}
```

#### Failure Response Sample

```
HTTP/1.1 401 UNAUTHORIZED
Content-Type: application/json
```

```
{
  "messages":{"code":"0100401"},
  "status":"error"
}
```

#### Token Response Raw Parameters

#	Parameter Name	Data Type	Description	Sample Value
1	access_token	String	The access token that is issued by the authorization server.	ee1073de-45d0-4040-b9c2-eddfa80280c0
2	token_type	String	The type of the token issued.	bearer
3	expires_in	int	The lifetime in seconds of the access token. For example, the value "32400" denotes that the access token will expire in nine hour from the time the response was generated.	32400
4	Scope	String	If identical to the scope requested by the client otherwise, REQUIRED.	api_scope
5	key	String	aes_secret_key and aes_iv collectively used to encrypt and decrypt further API request-response	
6	iv	String	aes_secret_key and aes_iv collectively used to encrypt and decrypt further API request-response	

Sample output of the decrypted **raw token payload** in JSON format:

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "access_token": "ee1073de-45d0-4040-b9c2-eddfa80280c0",
  "token_type": "bearer",
  "expires_in": "3600",
  "scope": "api_scope",
  "key": "aes_secret_key",
  "iv": "aes_iv"
}
```

Confidential



**POST /<version>/inquire/trd-act**

This API will allow members to inquire for trades & actions using API POST /<version>/inquire/trd-act endpoint. A maximum of 1,00,000 messages can be inquired in a single API call.

**Request**

## Request Header Parameters

#	Parameter Name	Data Type	Description	Sample Value
1	Authorization	String	Bearer token encrypted using the AES received as part of token response.  <access_token>	Basic MRZmwzdkje382jdw8ue93j dCaxH9rAM3hVIMJzFg==
2	nonce	String	A nonce uniquely identifies each server request. It should be a base64-encoded string in the format: ddMMyyyyHHmmssSSS:<6-digit random number>.	MjAwMTIwMTcxNjEyMjE0 TE6
3	consumerKey	String	The Member Consumer Key received as part of API Registration process.  <consKey>	consKey

## Request Body Payload (JSON)

#	Parameter Name	Data Type	Description	Sample Value
1	Version	String	API version	1.0
2	Data.msgId	String	Unique request number for each request <CODE><YYYYMMDD><nnnnnnnn>  Member / Custodian Code (Length: 4 or 5) • YYYYMMDD – Date format • nnnnnnnn – Running sequence no. starting from one i.e. For first request of the day, it should be (0000001).	XXXXX201310140000001
3	data.dataformat	String	Request data format: Response data format	CSV:CSV
4	data.trdactInquiry	CSV	Data Structure specified below	0,ALLTRDACT,,

## Trade Action Inquiry (trdactInquiry) Request Packet Structure (CSV)

Field Name	Description	Data Type	Size (in Bytes)	Sample
seqNo	The trade sequence should reflect the last trade sent by the server. For the first	INT	8	0

Confidential

Field Name	Description	Data Type	Size (in Bytes)	Sample
	request of the day, set it to 0.			
srchFilter	Search Filter	String	50	<ul style="list-style-type: none"> <li>• ALLTRDACT – Download all trades and actions</li> <li>• TMTRDACT – Download trades and actions of the Clearing Member as a Trading Member</li> <li>• CPTRDACT – Download only CP trades and related actions</li> <li>• ERRORACT-Download only all failure actions performed</li> </ul>
fill1	Filler	String	10	Leave it blank
fill2	Filler	String	10	Leave it blank

### Sample Request

#### Request Header:

```
POST /api/ncms-cm/trades-inquiry HTTP/1.1
Host: uat.connect2nscl.com
Authorization: Basic MRZmwzdkje382jdw8ue93jdCaxH9rAM3hVIMJzFg==
consumerKey: consKey
nonce: MjAwMTIwMTcxNjEyMjE1OTE6ODk0MjY3
Content-Type: application/json
```

#### Request Body:

```
{
  "data":
  "i3fJhLKZHhGdanX8csAP4sfqaXse/PO2ek84FMMocd8hLMVgHuOQREft6QsruHisVrqTBjDqAL4guyyVLLV3RnrYRRa3uuhRj+BdJI7UJE.....A6dy/yJaem0qa40X+5iUvteGpQ7BlpQ=="
}
```

Sample output of the decrypted **request body payload data** in JSON format:

The access token in the Authorization header, as well as the data parameter in the request body, are required to be AES-encrypted. When making an API call, the Base64-encoded string of these encrypted values must be used. Members should perform encryption using the AES secret key and IV provided at the time of token generation alongside the access token.

```
{
  "version": "1.0",
  "data":
  {
    "msgId": "00001201310140000001",
    "dataFormat": "CSV:CSV",
    "trdactInquiry": "0,TMTRDACT,,"
  }
}
```

Confidential

}

The access token in the authorization header, as well as the data parameter in the request body, are required to be AES-encrypted. When making an API call, the Base64-encoded string of these encrypted values must be transmitted. Members should perform encryption using the AES secret key and IV provided at the time of token generation alongside the access token.

### Response

Response Payload Structure (JSON)

#	Parameter Name	Data Type	Description	Sample Value
1	status	String	Response Status	success/error
2	messages.code	String	Refer to section “Message based response code”	01010000
3	data.msgId	String	Unique request number sent back in the response	XXXXX201310140000001
4	data.trdactinquiry	CSV	Data Structure specified below. Structure Separator “^”	Refer Sample Response

Trade Action Inquiry (trdactInquiry) Response Packet Structure (CSV – Structure Separator “^”)

Field Name	Description	Data Type	Size (in Bytes)	Sample
sysinfoResData	Control Record Structure	CSV	-	System Info Response Structure given below
trdResData	Detail Record Structure	CSV	-	Field Separator – “,” Record Separator – “^”  Trades Response Data Structure given below

Control Record Structure (sysinfoResData) (CSV)

Field Name	Description	Data Type	Size (in Bytes)	Sample
mktSts	Market Status  For list of Market Status please refer “Reference Codes” section	Short	2	6
currTrdDate	Current Trade Date (YYYYMMDD)	Long	8	20251028
sfill1	Filler	String	10	
sfill1	Filler	String	10	
maxSeqNo	Max sequence number sent in response	long	8	47
noOfRec	Count of trades sent in the response	int	4	2

Detail Record Structure (trdResData) (CSV) (Field Separator – “,” | Record Separator – “^”)

Confidential

#	Field Name	Description	Data Type	Size (in Bytes)	Sample
1	seqNo	Unique Sequence Number	Int	8	46
2	trdNo	Trade Number	Double	8	202504298000000001
3	actType	Activity Type. Refer Section "Reference Codes"	Short	2	2
4	trdPr	Trade price in paise	Float	4	12300.0
5	trdTime	Trade Time in jiffy format	Long	8	93824483328000
6	trdTmCd	TM code	String	5	XXXXX
7	trdTmBrnCd	TM Branch code	Short	2	2
8	trdStpType	STP type	String	1	5
9	trdStpNo	STP number	String	7	2025586
10	trdSecToken	Securities Token	String	11	757139
11	trdSecSymbl	Securities Symbol	String	10	ACC
12	trdSecSeries	Securities Series	String	2	EQ
13	ordNo	Order Number	Double	8	140000000000000001
14	Quantity	Trade Quantity	Int	4	110000
15	mktType	Market Type	String	1	N
16	cpCd	Cp code	String	12	XXXXXX00000001
17	cusCd	Custodian Code	String	5	XXXX
18	trdAcc	Client Code / Account No	String	20	10001
19	ordTime	Order Date Time	Date		10-OCT-25 09:22:33
20	transCd	Transaction Code. Refer Section "Reference Codes"	Short	2	6001
21	booktype	Book Type. Refer Section "Reference Codes"	Int	2	1
22	trdPan	Pan number	String	10	QAZZZZZZZZ
23	exchangeCd	Exchange Code	Short	2	1
24	trdUniqId	Trade Unique Id	String	20	757139202504298000000011
25	trdCmCd	CM code	String	14	XXXXXX
26	trdOrdUniqCd	Order Unique Code	Double	30	059001000000000000001811
27	actCliCd	Action Client Code	String	10	A1410353
28	actBy	Action taken by	String	12	XXXXXX
29	actDate	Action Date	Date		29-MAR-23
30	actTime	Action Time in jiffy format	Double	8	1372609300
31	actRcvdTime	Action Received Time in jiffy format	Double	8	1372010664
32	actErrCd	Action Response error code. Refer Sub Section "Async Response codes" in Section "Reference Codes".	Int	2	9
33	bsFlg	Buy Sell Flag. Refer Section "Reference Codes"	String	1	B
34	actId	Action Type. Refer Section "Reference Codes"	Short	2	1
35	Filler1	Filler	Double	8	
36	Filler2	Filler	Double	8	

Confidential

#	Field Name	Description	Data Type	Size (in Bytes)	Sample
37	Filler3	Filler	String	16	
38	Filler4	Filler	String	16	
39	Filler5	Filler	String	16	

## Sample Failure Response

**Wrong access token or expired access token**

HTTP/1.1 401 UNAUTHORIZED

Content-Type: application/json

```
{
  "messages":{"code":"0100401"},
  "status":"error"
}
```

**Error in encryption**

HTTP/1.1 400 BAD\_REQUEST

Content-Type: application/json

```
{
  "messages":{"code":"0100400"},
  "status":"error"
}
```

## Sample Success Response

The payload in the response to the API call, will be AES-encrypted string. The Base64-encoded string of this encrypted value will be transmitted. Members should perform decryption using the AES secret key and IV provided at the time of token generation alongside the access token.

**Actual Response**

HTTP/1.1 200 OK

Content-Type: application/json

```
{
  "status": "success",
  "messages":
  {
    "code": "01010000"
  },
  "data":
  "i1iwOPPNs0DJNSX8bswCpY65aWYSobTpBCR/UZAqacBiO8smQeHRa338+Ro7qGi8VOXSVzPCEP04oXWHd/AjOozKTge2vO9WiOJdJY3VJVhdcwZL3Gj4tEbXi4vxf+SfJ9bRxyfh5kMHZXvcInC55mpkHca9fdgm8pKmT0SdnQsKMJ11GMUYxKQtDvdzQXyxWI4GY3f"
}
```

**Response with Raw Data**

```
{
  "status": "success",
  "messages":
  {
    "code": "01010000"
  },
  "data":
  {
    "msgId": "XXXXX201310140000001",

```

Confidential

```

"trdactInquiry": "6,20251028,,,47,246,202504298000000001,2,12300.0,93824483328000,XXXXX,2,
5,2025586,757139,ACC,EQ,14000000000000001,110000,N,XXXXX0000001,XXXX,10001,10-OCT-25
09:22:33,6001,1,QAZZZZZZZ,1,757139202504298000000011,XXXXX,05900100000000000001811,
A1410353,XXXXX,29-MAR-23,1372609300,1372010664,9,B,1,,,,,47,
202504298000000001,2,12300.0,93824483328000,XXXXX,2,5,2025586,757139,ACC,EQ,1400000000
000001,110000,N,XXXXX0000001,XXXX,10001,10-OCT-25
09:22:33,6001,1,QAZZZZZZZ,1,757139202504298000000011,XXXXX,05900100000000000001811,
A1410353,XXXXX,29-MAR-23,1372609300,1372010664,9,S,1,,,,,"
}
}

```

## Appendix A - Reference Codes

### Market Type

Code	Description
1	Normal
2	Odd Lot
3	Spot
4	Auction
5	Call Auction 1
6	Call Auction 2

### Market Status

Code	Description
1	Preopen shutdown
2	Normal Market Preopen ended
3	Open Msg
4	Close Msg
5	Closing Start
6	Closing End

### Transaction Code

Code	Description
6001	Original Trade
5445	Trade Modification (Client Modification)
9001	All Actions (CP modification)

### Activity Type

Code	Description
1	Trade Modification (Client Modification)
2	Original Trade
5	CP Modifications

### Book Type

Code	Description
1	Regular Lot
2	Special Terms
3	Stop Loss / MIT

Confidential

Code	Description
4	Negotiated Trade
5	Odd Lot
6	Spot
7	Auction
11	Call Auction 1
12	Call Auction 2

**Client Type**

Code	Description
1	Cli
2	Pro

**Buy Sell Flag**

Code	Description
1	BUY
2	SELL

**Trade Status**

Code	Description
P	Pending

**Exchange Code**

Code	Description
1	NSE
2	BSE
3	MSE

**Action Type**

Code	Description
1	Original Trade
6	Buy Side Pre Image (Old CP)
7	Sell Side Pre Image (Old CP)
8	Buy Side Post Image (New CP)
9	Sell Side Post Image (New CP)

**Async Response Code**

Error	Error Code
SUCCESS	0
MODIFY ORDER DOES NOT EXISTS IN TRADES	1
ADMIN MODIFICATION – TRADE NO NOT NULL	2
ORDER MODIFICATION – MEMBER DISABLED	3
ORDER MODIFICATION – MEMBER ID MISSMATCH	4
ADMIN MODIFY - SEQUENCE NO NOT NULL	5
MODIFY ORDER - ALREADY CONFIRMED	6
MODIFY ORDER – INVALID NEW CUP CODE	7

Confidential

Error	Error Code
MODIFY ORDER – INVALID OLD CUP CODE	8
ORDER MODIFICATION WINDOW DISABLED	9
MODIFY CP - CLIENT CODE NOT NULL	11
MODIFY CP – NOT LATEST IMAGE	12
MODIFY CP – STPW_INST	13
MODIFY CP – CUP CODE NOT ACTIVE	14
MODIFY ORDER – BUY SELL FLG MISMATCH	15
MODIFY CP – OLD NEW CP CODE SAME	16
MODIFY CP – INCORRECT LENGTH	17
MODIFY CP – MARGIN INDICATOR	18
MODIFY CLIENT – CP CODE NOTNULL	21
MODIFY CLIENT – NOT LATEST IMAGE	22
MODIFY CLIENT – CLIENT CODE SAME	25
MOD_CLI_NULL	26
MODIFY CLIENT – INCORRECT LENGTH	27
MODIFY CLIENT – PRO TRADE	28
MODIFY CLIENT – MODIFY TO MEM CODE	29

\*\*\* End of Document \*\*\*

Confidential